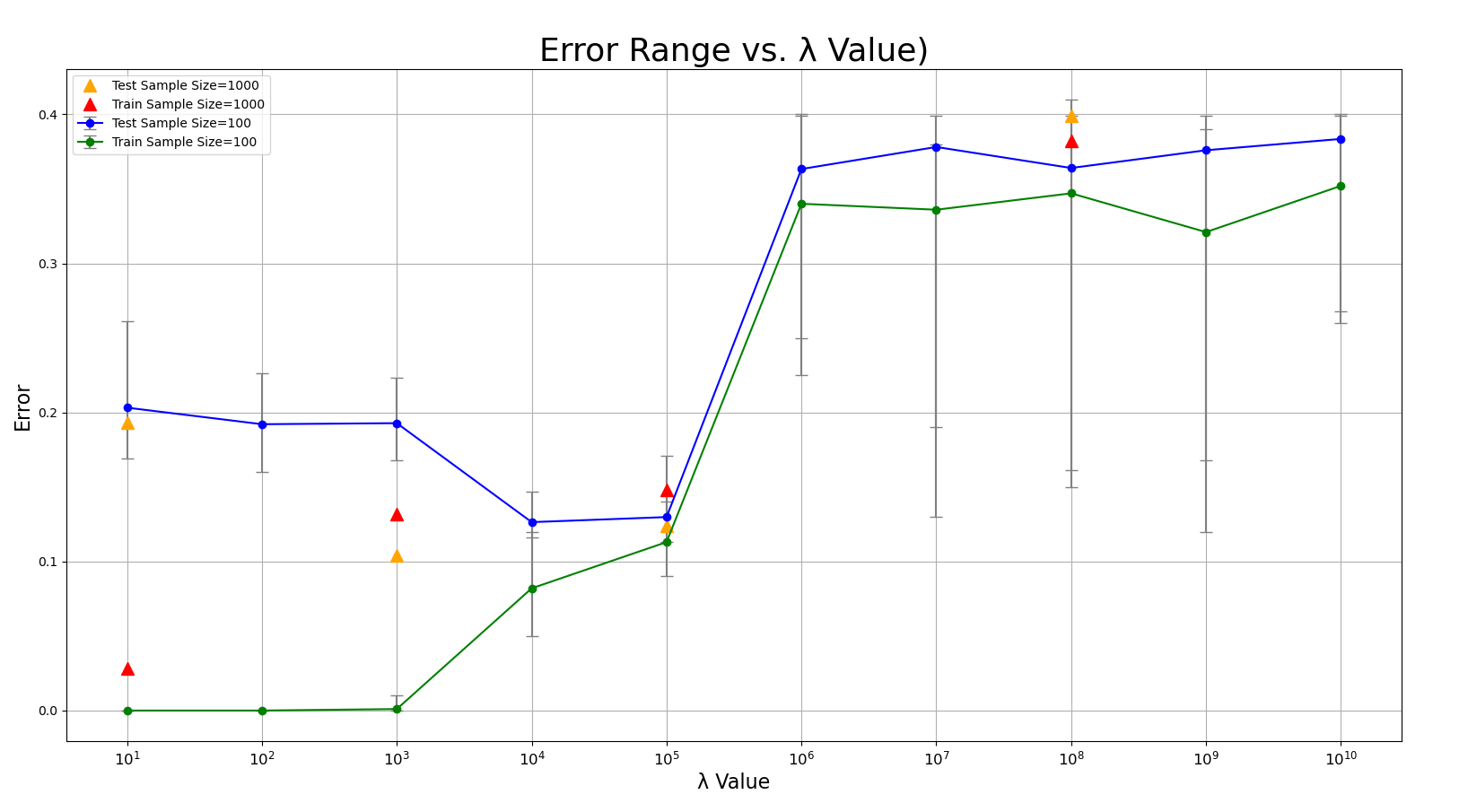
Intro To ML – Linear Predictors

Amit Zulan – 207299033, Dvir Ben Zikri - 315409508

1. Code is added separately.
2. (a) + (b)  
     
   (c)
   * + A smaller sample size should get a lower training error, less examples are easier to separate since they are relatively more scattered (on average), and smaller sample size increases the chances of the samples being linear independent, meaning they can be shattered (i.e. separable regardless of labels). Test error should be lower for higher sample size. Higher sample size avoids over-fitting and generalizes the separator. We can see both are reflected in the plot above for lower .
     + The trend of the training error should be increasing with since lower results in larger which causes a smaller margin, meaning we are trying to minimize the hinge loss on training sample, resulting in a separator that tries to minimize the number of samples inside the margin, and overfit the separator to the training sample. We indeed get and increasing train error with .
     + The trend of test error should not be increasing or decreasing. The trend should be of that convex function. For lower , as explained above, the separator tries to minimize the number of samples in its margin, overfitting it to the training sample, and not being general enough for the test sample. For larger , the separator minimizes its norm, resulting in larger margin, and the separator is decided by the further samples achieving high error in both train and test samples. The optimal solution should be received with that isn’t too small or too large. We indeed see this trend in the graph, for lower the test error is relatively high, while in larger it’s even higher, and minimum on the test error is received somewhere in the middle.
   1. Since we know that the rule/condition has at most 10 characters constructing the hypothesis class of might cause us to miss the best predictor since the sentence/condition might be of length 10 exactly or at least larger than the n chosen for .

Moreover for we increase the hypothesis class “wasting” computation time (although in the question we unlimited computational resources). Other than that, we might get a predictor which will get a zero loss on the S (training sample) but is not the predictor which get zero loss on D since its length is larger than 10 thus, we won’t get the optimal predictor.

* 1. Since is finite and we know that the doctor is using a condition of at most 10 characters that condition must be in thus D is realizable by and we can guarantee that any algorithm with training sample size m gets an error of at most ϵ, with a probability of at least 1 − δ over the random training samples if

In the question we demand

And since

* 1. Now that conditions with length larger than n and smaller or equal to 10 -> ,not included in we might not have the predictor used by the doctors hence D is not realizable by in this case we can use the PAC guarantee for the agnostic problem, the upper bound will be the lower bound for the largest H we can get in this case for :

1. We’ll find a given w that will shatter any combination of labels .  
   Instead of requiring , we’ll demand . Since , then .  
   Now we have d equations . Converting to matrix form we get , where .  
   Since U is a matrix of dimensions with exactly d rows consisting of independent vectors, U must be invertible. So, will separate every possible combination of

We define , therefore the optimal solution for is

And

We can see that if we multiply the l-2 norm squared by the dimension of w – d, we’ll get an upper bound on the l-1 norm squared, i.e.:

Let us prove:

Firstly, let’s look at the squared elements of i.e.:

Each one of them can be replaced by since absolute value don’t matter when we are squaring.

Then we shall look at the second part of the l-1 norm squared, i.e.:

We can see that:

This can be proven by:

Now since for each we have this element times we need times and the same goes for j.

Thus, all in all we get that we need one time for the squared element of l-1 norm squared and for the second element resulting in

And finally, we get:

Now since we can replace and solve with

Each solution we have for will hold for since (1).

Let us rewrite the minimization problem of the modified Soft SVM:

* 1. The vectors / matrices will be the same as the original Soft SVM problem expression but with a change in H – multiplication with d.
  2. We are required to prove .  
     We’ll divide into 3 cases:  
     (1)   
     (2)   
     (3)   
     Therefore, in either of the cases
  3. We need to prove .  
     We know , therefore Proof by induction:  
       
     Base case :  
     .  
     Explanation: Perceptron updates in each iteration with , since , and , then , i.e. only increases/decreases in increments of 1 (or doesn’t change). Since , then it must be at least 1.  
       
     Induction step assuming , proof for :  
     note: last equal is from geometric series sum.  
     We get Thus proving the theorem.
  4. Both previous sections were proven for any (in a) and any (in b), therefore we can take for section a, and for section b.  
     . Combining both:  
     .